

- ① Entities \rightarrow Core \rightarrow domains
 - ② DBcontext \rightarrow Application. \rightarrow This will be in Infrastructure \rightarrow Persistence.
Name \rightarrow ApplicationDbContext.cs.
 - ③ for migration \rightarrow use Persistence
in dropdown
 \Rightarrow add-migration "Init"-context ApplicationDbContext
 - ④ Update-database -context ApplicationDbContext
(if we want to add new entities)
 - ⑤ Persistence means it is been save and transient means temporary.
 - ⑥ IdentityDbContext : It provides basic tables.
 - ⑦ If we wanna add extra column \rightarrow it can be added in Infrastructure \rightarrow Identity
 \Rightarrow Models \rightarrow ApplicationUser \rightarrow IdentityUser
 - ⑧ Now And we have to add migration for above
 \Rightarrow add-migration "Init"-context IdentityDbContext
 \Rightarrow update-database -context IdentityDbContext
- g] Repository \rightarrow will be in \Rightarrow Infrastructure
 \Rightarrow persistence \rightarrow repositories for database operation. filename is "BaseRepository"
- h] There interface file will be in Core \rightarrow Application \rightarrow contracts \rightarrow persistence.
- i] 1) for Identity the services folder is present in Infrastructure \rightarrow Identity \rightarrow services.
(It is use for
- j] 2) There interface file will be in Core \rightarrow Application \rightarrow contracts \rightarrow Identity
- 13) contracts is use for defining the service which are available that is Identity, Persistence, Persistence.

15] Instead of modelbuilder we can use the folder ~~In~~ Infrastructure \Rightarrow Persistence. \Rightarrow configuration \Rightarrow It depends on Entities. it is use to modify the Entities (Ex: adhar card no. should be 10 length) so this validation will be there.

16] PersistenceService Registration, Application

In program.cs we have to add services so not to make the file lengthy we use such file.) individual. but we can't use such file for ~~so~~ Identity so we use IdentityServiceExtensions.cs.

17] connectionstring for Identity will also be present in IdentityServiceExtension and JWT Token configuration setting will be also present there.

JWT events also present in this.

18] Events mean Example. scrolling.

19] Program.cs :- All the file we made like InfrastructureRegistration is added in program.cs (line no. 67, 68, 69).

20] In Infrastructure \Rightarrow InfrastructureCore.

add general service like email, sms, otp (list to excel, list to database list to excel).

21] BlockStorage, cloudservice is added in service.egs. Infrastructure \Rightarrow Infrastructure.

22] Cache is also a type of service Infrastructure \Rightarrow Infrastructure \Rightarrow Cache. (Ex: temporary) (city example) (key is provided)

23] Core \Rightarrow Infrastructure \Rightarrow (DataAbstraction) It has only service (like supari demo)

24] Common column which are used in all entities \Rightarrow That is present in Core \Rightarrow Domain \Rightarrow Common \Rightarrow "AuditableEntity.cs" (Ex: createdby $\xrightarrow{\text{extend}}$ It is used in all entities and has implemented. you can check in entities.

25] The entire user need when request is performed the models are different and for response models are different

then we use Core \Rightarrow Application \Rightarrow models \Rightarrow Authentication

(Request and Response models are present in that) (Whene the alaq or dhkhanne ke alaq).

26] DTO : Data Transfer object.

27] DTO VM.

Transferring one object] when we want to be another object internally response to an user in UI then VM is used. (AppointmentBooking)

28] Events entities has various columns so we can show only particular columns so we create a DTO and use have called in Dto.

Core → Application → profile

29] Mapper → Auto-mapper, to mapping profiles, keep the key same in both the Direction matters.

CreateMap<Event, CreateEventCommand>().ReverseMap,

Source ↘
Source ↙ Destination.

ReverseMap → It can also for Destination to source

30] Mapper → CustomMapper → we can create by overriding auto-mapper to modify some auto-mapper filename "EventVmCustomMapper". when suppose keys are different.

Example : Under) we have to write 'extend' in Automapper

31] Automapper^{service} should be add in Application-Registration.cs

32] If have our own constants . suppose we have to add constant error msg so we have added in Core → Appn → Helper → ApplicationConstants.cs . (We have there Key we have to use that key Ex:-("en"))

33] To extend any method or modify we use Core → Appn → Helper → Encryption.

34] Exception → Core → Application → Exception.

API:-

4] Utility:- To maintain the file size and file type.

1] Swagger helper :- Swagger UI

2) Middleware :- It is use to global exception handling.

When request hit till the time it comes to controller that period is middleware.

Middleware services are add in

MiddlewareExtensions.cs of ExceptionHandle and their service of middlewareExtensions are added in program.cs.

3) Extensions :- Controller has Authorization in API but we cannot add token there so

to add the key and configure we use ServiceExtensions.cs.

• HealthCheckExtension:- It is a package which is use to check the health API.

• AppExtension :- service or configuration are added and it is called in program.cs.

⑦ Validator :- It is use to add validation.

⑧ serilog :- it is use for log

⑨ loggedin \Rightarrow to read from services.